



Case for Deploying Complex Systems Utilizing Commodity Components

Barry S. Bryant

barry.s7.bryant@lmco.com

Phone 256-544-7404

R. Lee Pitts

robert.l7.pitts@lmco.com

Phone 256-544-0666

George Ritter

george.ritter@ums.msfc.nasa.gov

Phone 256-544-8269

Marshall Space Flight Center

Lockheed Martin Space Operations



Introduction



Introduction

- The primary focus of this presentation is the transition of a Space Station ground system from a client/server UNIX based system to a client/server system based on commodity priced and open system components
- In covering this transition, the presentation will discuss
 - ◆ A definition of the **HOSC Ground System** and its capabilities in order to lay the ground work for the transition
 - ◆ The reasons why the transition was necessary in **Motivation for Change**
 - Identifies many of the areas to look for when considering changes to your system
 - ◆ Several methodologies or **Options** that were considered once the decision was made that some change was required
 - Options considered ranged from continuing on in the same vein with one-for-one replacement to a complete re-baseline
 - ◆ The **Goals** that were identified early in the transition process
 - Documents what was to be accomplished and added restrictions for impacts to operations during the transition



Introduction

- In covering this transition, the presentation will discuss
 - ♦ The **Implementation** of the goals
 - Details the transition of the HOSC
 - ♦ The primary **Initiatives** that were identified, approved and implemented as part of the transition
 - ♦ The methods used to identify, define, gain approval and implement the initiatives in **Conclusions**



HOSC Ground Systems



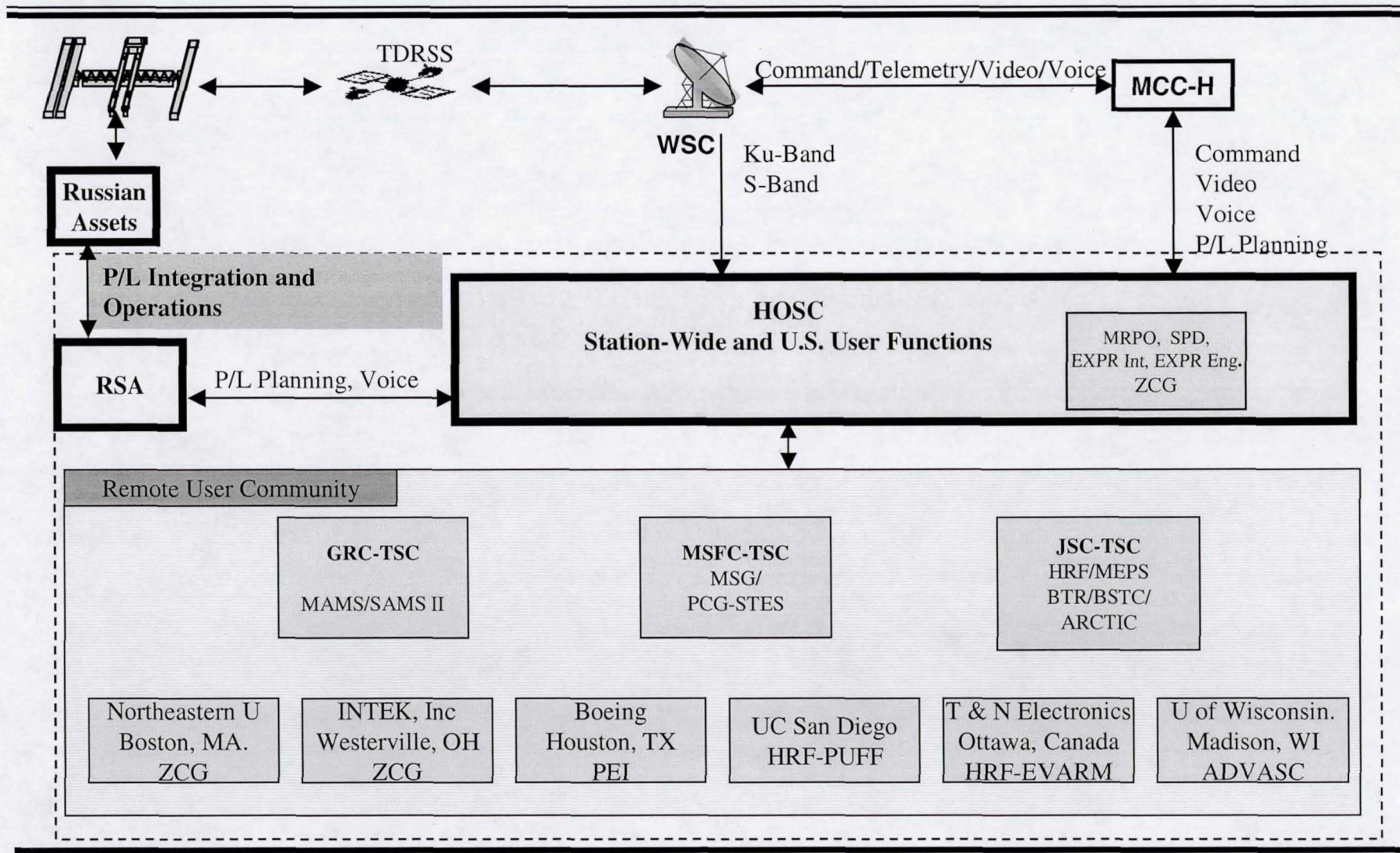
HOSC Ground Systems

- The HOSC hosts the ground systems for all payloads in the US portion of the International Space Station
 - ♦ The HOSC is a multi-mission facility
 - ISS Operations supports a diverse user base of payload investigators
 - Facility class payloads and individual experiments supported simultaneously
 - ♦ HOSC ground systems are accessed by users from around the world
 - All payload users utilize web services in support of operational activities
 - All science data are distributed from the HOSC
 - Realtime components allow users to command, manage payloads, and process health and status telemetry
 - ♦ Payload systems managers (Cadre) are located at the HOSC
 - ♦ Payload users may be located locally or remotely with all services available



HOSC Ground Systems

Increment 5 Interfaces





HOSC Ground Systems

- NASA offers services through 4 primary systems
 - ◆ Major systems in the HOSC architecture are
 - Enhanced HOSC System (EHS)
 - Payload Data Services System (PDSS)
 - Payload Planning System (PPS)
 - Telescience Resource Kit (TReK)
 - ◆ A number of auxiliary services are available as well
 - ◆ Traditional Voice Distribution System
 - ◆ Internet Voice Distribution System (IVoDS)
 - ◆ Video Distribution System
 - ◆ HOSC systems are online locally at KSC to support ISS payload test and integration
 - ◆ The HOSC also supports STS launch activities
 - ◆ A primary HOSC component (EHS) serves as the ground system for the Chandra X-Ray Observatory in Cambridge, Mass.
 - ◆ The HOSC systems are designed to provide support from single users up to large facilities
-



HOSC Ground Systems

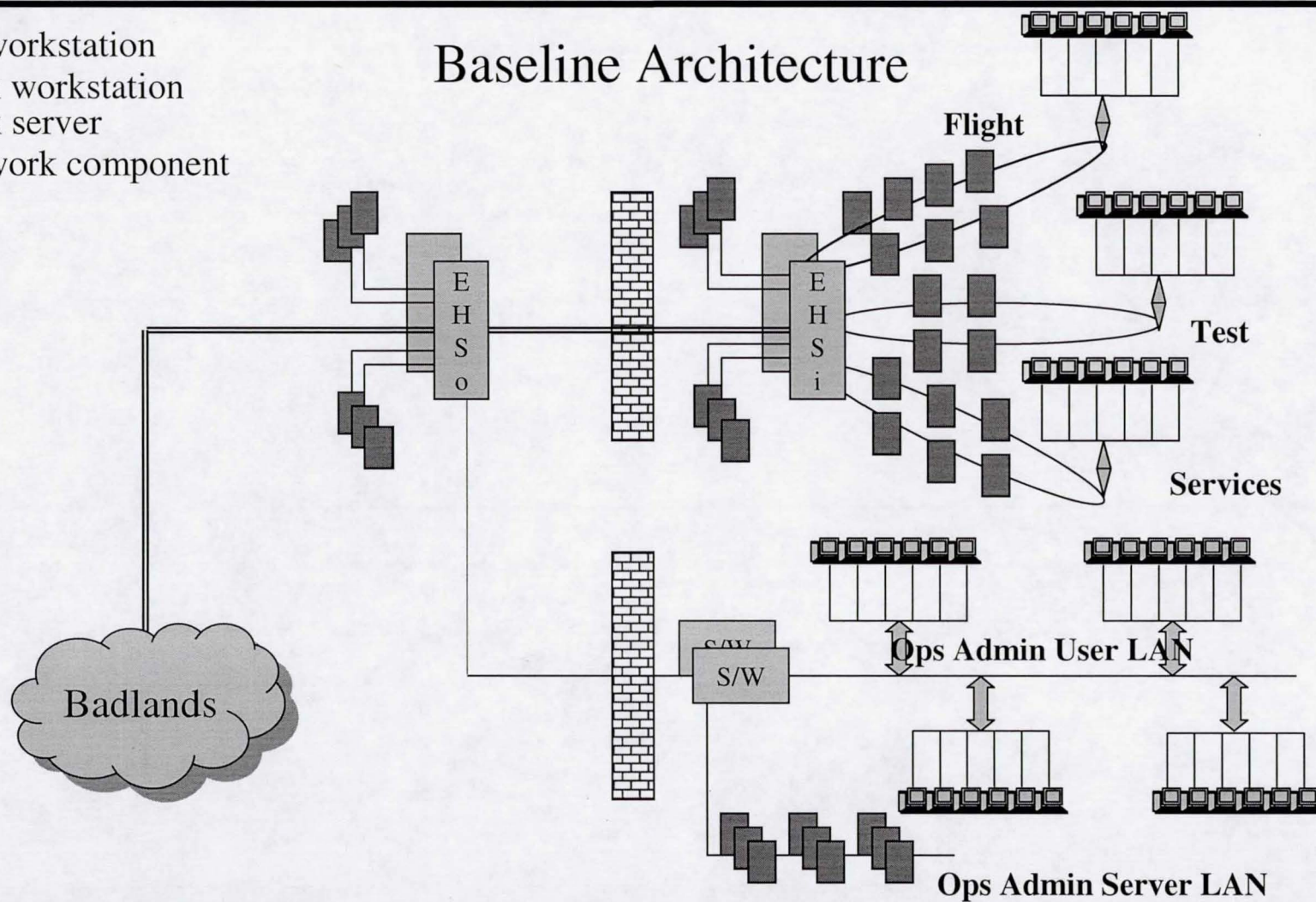
- The systems are partitioned into a Private Domain and a Public Domain
 - ◆ Private Domain is secured by redundant Firewall and DNS servers
 - ◆ Public Domain is secured by router access lists and firewalls
 - ◆ Network traffic between the Private Domain and the Public Domain is secured through protocol controls
 - ◆ HOSC Mission LANs reside in the Private Domain
- HOSC LANS are partitioned by function with routers providing OSI/ISO layer 3 separation
 - ◆ Mission and test LANs are isolated
 - ◆ User Operations Area LANs for Users are in the Public Domain
 - ◆ Public DMZ LAN is outside the Firewall for necessary but sacrificial systems
 - ◆ A separate and isolated Ops Admin LAN is provided for Cadre users



HOSC Ground Systems

- PC workstation
- Unix workstation
- Unix server
- Network component

Baseline Architecture





HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ◆ Multi-tasking server environment developed and executing on a UNIX OS
 - ISO/OSI compliant communications stack implemented to approved and established standards
 - EHS applications are built to ANSI C language standards and are POSIX compliant
 - Database applications use a modern DBMS with the data presentation layer supported by a standard SQL interface
 - Standards based Data Presentation Layer for
 - EHS WEB based applications
 - X-windows protocol for EHS X-Windows based applications
 - Security for access control is based on user profiles/roles
 - System-wide Monitor and Control and Network Management functions
 - Failover capability for all EHS critical components
 - ◆ Homogeneous client-server UNIX workstation environment



HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ◆ EHS architectural hardware components include
 - Telemetry Servers
 - System Management Servers
 - Internal/External WEB servers
 - Non Realtime Development Servers
 - PIMS/Timeliner Servers
 - Network Management Servers
 - NTP Servers
 - Drop Boxes
 - Command Servers
 - ERIS Servers
 - DNS/Print Servers
 - Test Servers
 - Database Servers
 - Firewall Platform
 - EHS Workstations
 - FDDI LANs, Ethernet LANs, routers, HUBs, switches, and serial data/clock interfaces



HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ♦ EHS ground system services are
 - Command Processing
 - Provides users the capability to view, modify, initiate, and status commands
 - Provides unique controls and security functions necessary to safely execute hazardous and critical command functions for a variety of missions
 - Provides programmatic interfaces to remote users and other EHS applications
 - Data Acquisition & Distribution
 - Acquires telemetry data from multiple sources
 - Error checks and generates data quality information
 - Encapsulates data into standard EHS packets for distribution
 - Database Services
 - Converts inputs into EHS ISS Command and Telemetry Database
 - Validates, produces, and delivers command and telemetry databases
 - Provides DBMS monitoring for status and performance
 - Provides a centralized Database Change Request (DCR) system



HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ♦ EHS ground system services are
 - Operations Control Mission Software
 - Collection of software tools on the user workstation used to manage payload operations for ISS payloads
 - Creates automated procedures which work in conjunction with the custom Timeliner software to generate ISS unique command packages
 - Programmatic access to EHS capabilities allows for tight situational awareness
 - Payload Information Management System
 - Provides a workflow engine with user notifications and status updates
 - Contains a data vault for the storage and management of operational procedures, documents, and change requests
 - Acts as the operational hub for payload users to interchange flight items with the cadre
 - System Monitor & Control
 - Centralized configuration, control, and monitoring of hardware and software elements
 - Performance and event data logging and reporting
-



HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ♦ EHS ground system services are
 - System Services
 - Standards compliant libraries for portability enhancement
 - Centralized User Profiles/Roles defining user privileges that is distributed by Systems Services to other EHS applications
 - Enhanced security services to applications above O/S minimums
 - Telemetry Processing
 - Traditional telemetry and data routing services for packet data
 - Logging, management, retrieval, and playback of processed data
 - Ability for a user to build and use graphical, textual, and tabular displays in realtime
 - Ability for a user to build and use scripts in realtime for the command, control, and analysis of all EHS monitored components
 - Delivery of telemetry data to local and remote users



HOSC Ground Systems

- EHS is the ISS ground system utilized primarily by the HOSC Cadre (payload managers) located at the HOSC
 - ♦ EHS ground system services are
 - Utilities
 - Common messaging and user interface functions on all EHS platforms
 - File services for managing files on local EHS workstations
 - Bulk validation services that check user products for valid MSID references and proper syntax and command mnemonics
 - Print services, timing display services, and workstation environment customization
 - Web Infrastructure
 - Provides a secure interface for select non-realtime services
 - Utilizes standards based VPN technology
 - Services accessible based on user privileges, project, and activity

EHS is the gateway ground system for users of the International Space Station!



HOSC Ground Systems

- PDSS is the ground distribution system for ISS science data
 - ◆ PDSS receives input data as
 - 192 kbps S - Band ISS Realtime data stream
 - 36 packets per second
 - 50 (soon to be 150) Mbps Ku - Band ISS Realtime data stream
 - Approximately 8,000 CADU per second
 - Up to 82,000 CCSDS packets per second
 - ◆ Encapsulates CCSDS packets and BPDUs in EHS headers for further processing
 - ◆ Generates and distributes data stream statistics
 - ◆ Multithreaded UNIX server environment
 - Designed for high availability
 - No direct user access
 - ◆ Highly available storage for up to 24 months of user science data

PDSS provides a standard delivery method for science data to users of the ISS!



HOSC Ground Systems

- PPS automates the planning, scheduling, and integration of payload operations during pre-increment planning, weekly planning, and real-time execution.
 - ◆ Multi-tasking server environment composed of heterogeneous hardware and software platforms
 - ISO/OSI compliant communications stack implemented to approved and established standards
 - Applications are built to ANSI C language standards
 - Database applications use a modern DBMS with the data presentation layer supported by a standard SQL interface
 - Standards based Data Presentation Layer for
 - WEB based applications
 - X-windows protocol for X-Windows based applications
 - ◆ PPS is a stand-alone system for short term, intermediate, and long term payload planning tools



HOSC Ground Systems

- PPS automates the planning, scheduling, and integration of payload operations during pre-increment planning, weekly planning, and real-time execution
 - ◆ PPS is composed of several unique tools which operate on various platforms
 - Some tools were developed locally and reused extant hardware
 - Some tools were imported to provide a “same-as” environment
 - Items were integrated on site to provide an integrated system utilizing web and X-window technology
 - ◆ PPS is composed of three (3) primary software tools
 - Reports – Access portal for review of data flow plans
 - User Requirements Collection (URC) - Web based capability to create payload resource definitions to determine resource utilization and schedule planning for various ISS operational modes
 - Data Systems Routing and Configuration (DSRC) - Used to model TDRSS and ISS resources and scheduling constraints for data, voice, and video communications

PPS is integral in the scheduling of scarce ISS resources!



HOSC Ground Systems

- TReK is the “Every man’s” ground system
 - ◆ It is for the individual or small payload team who wants autonomy
 - It provides command and telemetry processing and interface to the HOSC via the EHS remote command interface
 - TReK was designed with the science investigator in mind
 - ◆ TReK is unique for the HOSC in that it is completely self contained
 - Local database, playback and recording capability provided
 - An Application Programmatic Interface (API) is available for customization

TReK is the final HOSC component which allows all users of the ISS affordable access to science



Motivation for Change



Motivation for Change

Overall Needs

- A number of issues needed to be addressed at the system level
 - ◆ Serial data interfaces are becoming obsolete
 - Data and voice are common over the internet
 - Spacelab vintage hardware is becoming increasingly hard to spare
 - ◆ Budget pressures are affecting space and science
 - Science wants more money for science and less to “operate” science
 - Congressional input in the budget process sometimes overrides operational imperatives
 - ◆ O&M recurring costs are often the largest part of the budget for long duration projects like the ISS
 - ◆ Early procured hardware is being discontinued and is hard to spare and maintain
 - ◆ As the HOSC refurbishes, gains can be reaped by the insertion of new technology
 - Use of a consolidated storage architecture such as SAN or NAS
 - Retirement of sunset technology such as FDDI
 - Increased compute capability
 - New system management capabilities
 - Incorporation of open source items



Motivation for Change

Success

- The HOSC has been a success and is vigorously supporting the ISS
 - ♦ Success brought in new users and more opportunity
 - ♦ ISS expansion has increased the number of HOSC users and the way the HOSC services are utilized
 - ♦ Payload users want to conduct operations at home
 - Familiar environment
 - Lower overall cost
 - May want a limited copy of the HOSC
 - ♦ More users desire access to ISS related data; i.e. collaboration, schools
- These opportunities are good but puts stress on the current system and points to new needs of the community
 - ♦ Lower cost platforms
 - ♦ More flexible platforms operationally to meet the needs of more users
 - ♦ Ability to support a diverse number of project types



Motivation for Change

Hardware Drivers

- Primary platforms (servers and workstations) were reaching the End-of-Life
 - ♦ Some in excess of 5 years old
 - Workstation EOL in 12 months (when started)
 - Server EOL in 24 months (when started)
 - ♦ Platforms could not be expanded
 - ♦ Some platforms are from the program initial start-up
 - ♦ Delays in deployment due to ISS delays and normal program deployment
 - ♦ Buys were made early in development/deployment cycle
 - Did not maximize the performance per dollar for deployment
- Waning/non-responsive support for COTS products on the primary vendor
 - ♦ Drives higher ground system (EHS) application software maintenance cost
 - ♦ Limits the applications that can be hosted on the workstation (products lag behind)
- Concern over long term vendor viability
 - ♦ Aggressive computer market is slowly winnowing underachievers
 - ♦ Market crash accelerated vendor difficulties
 - ♦ A specific vendor had major losses



Motivation for Change

Hardware Drivers

- High cost per seat for replacement/maintenance
 - ◆ Over 130 workstations
 - ◆ Over 75 servers from low-end single processor to high-end SMP
- Flight users require multiple platform types
 - ◆ Usually UNIX and Windows
 - ◆ Increases development, test, maintenance, training and licensing costs
 - ◆ Some ISS products are only supported on the PC
- Other vendors platforms are in various stages of obsolescence
 - ◆ This is a normal part of being a multi-project and mission host facility
 - ◆ Need a solution which supports normal progression of obsolescence
 - ◆ Includes a variety of hardware
 - Servers
 - Workstations
 - Special purpose hardware
 - Peripherals



Motivation for Change

Budget Crisis

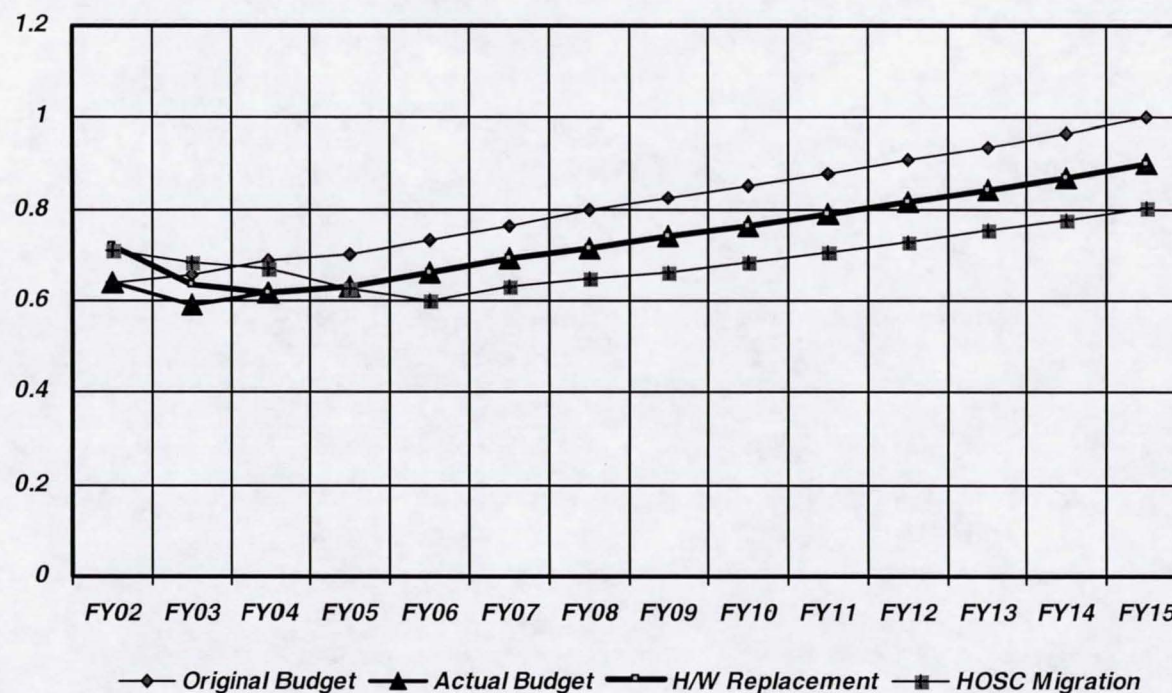
- ISS overruns in some areas required cut-backs across many ISS elements
 - ◆ Independent of the competencies or successes
 - ◆ Program wide review of budget
- Projected budget would increase by 42 % by 2012 with no refurbishment
 - ◆ Government budget reduction would reduce total budget growth by 19% with no refurbishment
 - ◆ Refurbishment would not be funded
- A onetime hardware refurbishment would increase the total growth to 91% of the original budget with little discretionary money
 - ◆ Refurbishment costs are in the first two (2) years
 - ◆ Cost would continue to rise in parallel with earlier predictions
 - ◆ New requirements would be costed separately



Motivation for Change

Budget Crisis

Normalized Cost Comparison





Motivation for Change

Budget Crisis

- Proposals were detailed and prototyped based on commodity items and open systems
 - ◆ Promote a cost model which is sustainable
 - ◆ Server consolidation where practical
 - ◆ Reassessment and decoupling of COTS where value is questionable
 - ◆ Use of commodity based hardware and software platforms ubiquitously
 - ◆ Technology injection to reduce recurring cost, increase user satisfaction, increase availability, and simplify operations
 - It was/is hoped that the proposal would embody the Quicker, Better, and Cheaper paradigm
 - ◆ By consolidating servers, less platforms will have to be managed
 - ◆ By building refurbishment into the model, our systems will not become obsolete
 - ◆ By using commodity platforms, market forces will keep cost down
 - ◆ By using commodity platforms, users will be familiar with the interfaces
 - ◆ Wise use of COTS will reduce recurring cost and increase satisfaction with COTS
 - ◆ New technologies may significantly reduce cost during the refurbishment
-



Motivation for Change

COTS

- COTS packages contribute to an ever increasing cost spiral
 - ♦ Many COTS are underutilized
 - ♦ COTS have a life of their own and often are renewed beyond their need
 - ♦ Platforms achieve performance increases by utilizing multi-processing
 - COTS are often licensed by CPU
 - Older platforms have more CPUs to meet requirements versus new technology
 - ♦ Increases in user base multiplies number of platforms
- Integrating COTS is expensive
 - ♦ COTS and O/S versions must be complementary
 - O/S upgrades must be applied as needed for security and operational needs
 - Inconsistencies must be reconciled with each upgrade
 - ♦ Vendor support may wane or force jumps in versions to support capabilities
 - ♦ Some vendors use proprietary methods which may not be interoperable
- In 1999, the HOSC had over 60 COTS products



Motivation for Change

Stability

- The HOSC has a stable and generic requirements base
 - ◆ ISS and Chandra programs are supported on-orbit with solid capabilities
 - ◆ Application code was developed to encompass general operational capabilities
 - Experience has shown which areas are in need of process improvement
 - Operational environment and utilization has identified needed performance enhancements
 - ◆ Redundant or obsolete capabilities and features have been identified
 - ◆ New capabilities and features can be incorporated and will reduce cost while enhancing operability
 - Highly available storage
 - Fault tolerant systems
 - Consolidation of COTS products which accomplish similar tasks
 - Introduction of high performance network hardware
 - Introduction CISC servers (Intel type processors)



Options



Options: Continue On

- Replace servers and workstations one for one
 - ◆ Start with workstations
 - Reach EOL sooner
 - Largest deficiency in performance
 - ◆ Follow up with servers as budget and workload allows
 - ◆ Work special purpose systems as required
 - Address VME devices
 - Focus on platforms with limited/waning COTS support
 - ◆ Replace systems without incorporating operational lessons learned
 - Users need more screen real estate
 - New tools may not be supported in operations
 - Enhance performance based on actual operational usage
 - ◆ Risk to user services is minimal
 - Primary effort is to replace aging hardware
 - Minimal software changes required

One for one replacement would have cost nearly as much as the initial outfitting



Options: Consolidate Servers

- Replace servers with the high cost, higher performance platforms
 - ◆ Modify software to allow for multiple server functions to run on a single platform
 - ◆ Incorporate limited lessons learned from operational use
 - ◆ Minimizes the number of replacement servers required
 - ◆ Risk to user services is higher
 - More software is affected by the change
 - Increases operational complexity
 - ◆ Adding the capability for consolidation of servers would not significantly reduce the replacement costs
 - Larger servers cost more to purchase and maintain
 - Due to operational and design constraints, only a few server functions could be consolidated
 - ◆ This is a variant of the one for one replacement with slightly more risk

Provides some long term savings while addressing a few critical operational needs



Options: Complete Re-Baseline

- Other options considered for refurbishing the HOSC included a complete paradigm shift
 - ♦ One option considered was to transition to an entirely PC/W2K environment
 - Required both server and workstation UNIX software to be rewritten to run on a PC
 - Opened the door for concerns over security and the multi-tasking capability of the PC
 - Required a large learning curve for a complete team of UNIX developers, system managers and maintenance personnel
 - The risk to the user is enormous with this option since a large percentage of the system would have to be re-written, re-integrated and retested
 - The magnitude of the software development would negate much of the gains in hardware savings and the users would have to be retrained and re-certified under a different model
 - ♦ Another option considered was a mainframe architecture with dumb clients
 - Required very expensive mainframe servers and a major rewrite of the software
 - This option was quickly ruled out as cost prohibitive

A major paradigm shift was beyond budgetary and philosophical scope



Options: Migrate

- Migrate the HOSC systems incrementally
 - ◆ Maintain some old
 - Migrate to like systems, incrementally
 - Preserve the large investment in user products
 - Minimize changes to systems that are performing at acceptable levels
 - Minimize perturbations to external interfaces
 - ◆ Integrate some new
 - Incorporate high value items that support the needs of the user base
 - Consolidate functions where there is an obvious return on investment
 - Re-evaluate all COTS and isolate or eliminate when possible
 - Evaluate commodity based platforms through prototyping and user testing
 - Incorporate commodity systems into the operational environment while maintaining user access to legacy systems
 - Evaluate open O/S and COTS through extensive prototyping

Migration allows the incremental upgrades while preserving stable user interfaces



Options: Migrate

- The risk to the user is minimized as new capabilities are added incrementally and access to legacy capability is maintained until the users are comfortable with the new method
- The migration option offers the greatest return on investment with acceptable risk and upfront costs
 - ♦ Minimize re-writing software except where the value is the greatest
 - ♦ Take advantage of software developed to standards and port to cheaper OS/platforms
 - ♦ By taking this route, the software modifications are paid for by the savings in hardware



Options: Technology Insertion

- Selectively replace pieces of the system where new technologies provide a large advantage over the current capabilities (applies to all baseline options)
 - ♦ Upgrade networks
 - Consolidate where possible
 - Increase performance and flexibility
 - Move to lower maintenance systems
 - Replace sunset technologies
 - Minimize risk by putting the new system in place and incrementally moving platforms to the new system as they are replaced
 - ♦ Migrate firewalls
 - Replace systems with higher performance platforms
 - Integrate/consolidate multiple firewalls
 - Transition from proxy based to level three filtering based
 - Minimize risk by transitioning one firewall to the new system and incrementally consolidating other firewalls to the new system



Options: Technology Insertion

- ♦ Migrate from prime/backup RAID storage to central storage
 - Provide access to all data from any server
 - Eliminate the need to transfer data as systems migrate from development to test to simulations to operations
 - Integrate both SAN and NAS capabilities based on the access methods of systems utilizing the central storage
 - Create a heterogeneous environment integrated by open standards
 - Minimize risk by moving pieces of the system incrementally as manpower allows for the software to be modified to take advantage of the central storage



Goals



Goals

- The HOSC is a multifaceted facility which is critical to the success of the ISS
- However a number of items were forcing change on the HOSC
 - ♦ Systems needed replacement
 - Network Hardware
 - Server and workstation hardware
 - Firewall
 - COTS packages were becoming more restrictive to platforms
 - ♦ Increasing spiral of COTS costs
 - Pricing of some packages were increasing dramatically
 - Versions updates of some packages required re-engineering
 - Some packages offered minimal enhancements to capabilities
 - ♦ Severe ISS budgetary pressure in excess of \$5,000,000,000.00

We had to do something to preserve the HOSC and SCIENCE!



Goals

- The Good News
 - ♦ The HOSC was standards based
 - All endeavors were standards based to support portability
 - Current commodity offerings had expanded greatly to include most standards
 - ♦ The HOSC has a stable requirements base and the mission is well defined, though evolving
 - ♦ Moore's law and open systems advances offered new opportunities
 - ♦ We have a lot of experience and a knowledgeable support group
 - ♦ We have a customer who is willing to try something new and accept a reasonable level of risk



Goals

- The Bad News
 - ♦ Timelines were/are short
 - ♦ Money is tight and justification is required for everything
 - ♦ Many people had to realign their thinking
 - ♦ Little or no disruption of ongoing operations
- Areas investigated were not restricted to any one area or method
 - ♦ Decoupling COTS with custom code
 - ♦ Reducing the number of COTS products
 - ♦ Evaluating new types of platforms and moving to commodity based systems
 - ♦ KISS – Keeping It Simple



Goals

- The goals were and are still high
 - ◆ Impact to ongoing operations and scheduled activities must be minimal
 - HOSC Cadre is involved at all levels, from Change Request (CR) to transition
 - Being the payload system managers, the Cadre supports onsite, 24x7
 - The Cadre is the primary user of HOSC platforms and EHS/PDSS software
 - The Cadre exercises most capabilities and has a substantial investment in user products
 - The Cadre interfaces with station crew and ground users
 - The Cadre is our primary customer therefore impacts at all levels must be minimized
 - Reuse of user products (displays, script, computations, etc)
 - User interface must be maintained or a more natural mechanism utilized
 - Test and verification of new or transitioning capabilities must occur in parallel
 - Transitions of capabilities must occur with little or no interface impacts

Operational users have limited resources or flexibility to accommodate change



Goals

- The goals were and are still high
 - ◆ Impact to ongoing operations and scheduled activities must be minimal
 - The HOSC has an overall availability record of 99+% that must not be compromised
 - Current numbers for availability are for 98%
 - Highest availabilities are on core services (TLM, CMD)
 - Near continuous availability is required on infrastructure items such as firewalls, networks, and storage
 - New systems and capabilities must exist and transition in parallel
 - Increased reliability, maintainability, and availability is a migration goal
 - New technologies will be inserted to enhance availability
 - Lowers impact on O&M as the user base increases
 - Reduce and simplify the operations and maintenance of the HOSC

Uninterrupted operations must continue



Goals

- The goals were and are still high
 - ◆ Impact to ongoing operations and scheduled activities must be minimal
 - The HOSC has International and National partners
 - Users are widely dispersed geographically
 - ICDs with ESA, NASDA, CSA, RSA, ASI and CONUS partners (TSCs)
 - Only a small subset of users are at the HOSC
 - A generic user interface document for overall capabilities is provided
 - These documents take years to negotiate and finalize
 - ISS user (local and remote) services and protocols must be maintained and observed
 - Users extensive capabilities when operating either locally or remotely must be preserved by agreement
 - EHS desktop with identical capabilities (subject to privilege) of the POIF
 - Web desktop with identical capabilities (subject to privilege) of the POIF
 - Command interface from remotes to the HOSC

Certification of Flight Readiness (CoFR) will be observed



Goals

- The goals were and are still high
 - ◆ Impact to ongoing operations and scheduled activities must be minimal
 - The HOSC security model will be maintained and extended
 - The HOSC is IP based and subject to a broad range of attacks
 - The HOSC has tightly regulated internet access
 - A wide variety of counter-measures are employed
 - Security model is broad based and relies on personnel, architectural, and software measures
 - A compromise of security could jeopardize not only the HOSC but any interfaces
 - International Space Station and payloads
 - ESA, NASDA, CSA, RSA, ASI and CONUS partners (TSCs)
 - Johnson Space Flight Center

Disruption of security could be catastrophic for the ISS program



Implementation



Implementation

- As stated earlier the HOSC had become a heterogeneous computing environment between the various systems
- A concerted attempt was made to identify a common platform to host as many systems as possible
 - ♦ Provide user satisfaction
 - ♦ Reuse of skills and methods
 - ♦ Availability of tools and training
 - ♦ Availability of COTS packages
 - ♦ Cost
- Since the HOSC primary O/S type is UNIX, Linux was selected on the server side
 - ♦ Reuse of skills and methods
 - ♦ Multi-vendor support
 - ♦ Wide array of COTS available including the HOSC current suite
 - ♦ Excellent cost advantage



Implementation

- The HOSC has been required to support multiple clients on multiple vendors
 - ◆ Common items of many are the X-window protocol and telnet access
 - ◆ Remote users and Cadre members have a large quantity of Windows PC supported tools
 - COTS tools based on the home and office environments
 - Custom software built on the PC due to the inexpensive environment and large quantity of development tools
 - High performance of graphical applications
 - Unlimited expansion capability
 - ◆ Windows 2000 (NT was originally prototyped) PC was selected on the client side as the single platform
 - Client apps are being moved from X-window to native PC one at a time
 - A COTS X-window server provides access to legacy applications
 - HOSC built an integrated Launchpad for the PC to provide a common interface for the user



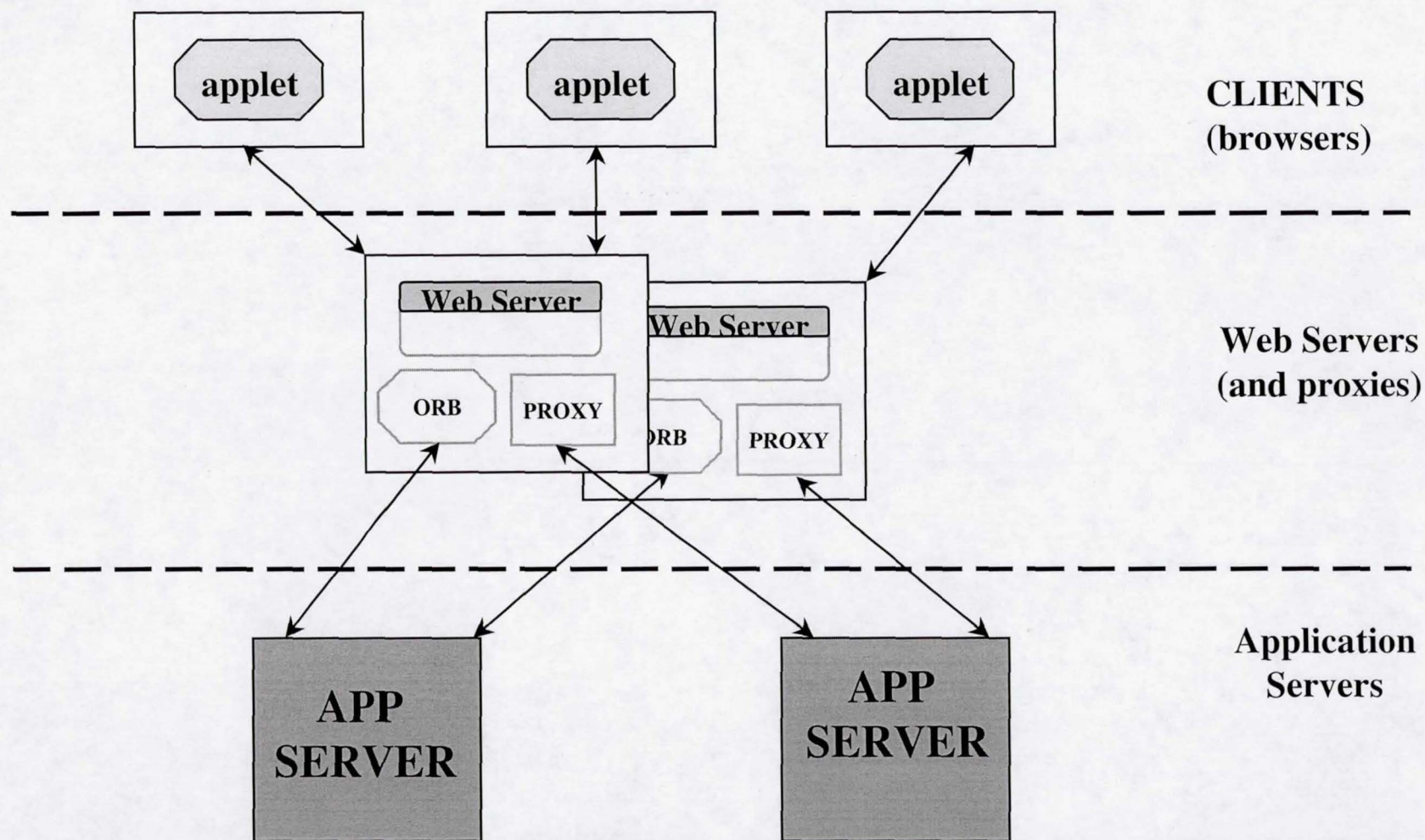
Implementation

- These decisions on client and server components in the architecture and hardware required a few ground rules
 - ◆ First, the clients can not be allowed to interfere in server processing either directly or indirectly
 - ◆ Second, users must not be allowed to interfere with other users
 - ◆ Third, the architecture must be transitioned in phases
 - ◆ Finally, internal and remote users must be supported transparently
- Fortunately, the HOSC had developed a multi-tiered architecture in the late '90s to support its secure web interfaces
 - ◆ Client side is hosted on a workstation or PC (no unique client other than JAR files)
 - ◆ Web servers acting as proxies for HOSC services with no persistent data or security files (passwords, etc)
 - ◆ HOSC services supported by database, application, and security servers completely internal



Implementation

Web Three Tier Architecture

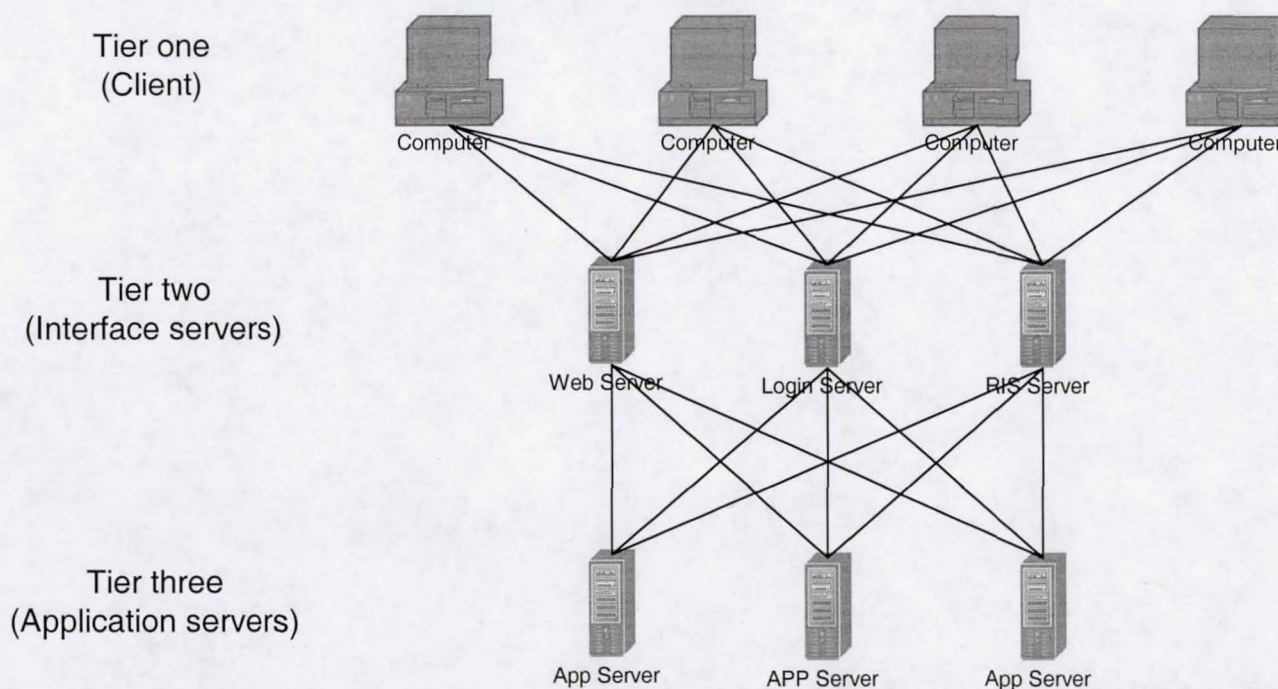




Implementation

EHS Three Tier Architecture

- The HOSC extended the web three tiered architecture to EHS systems
 - ◆ This allowed the migration of capabilities independent of the platform types
 - ◆ User workstations were almost immediately replaced with PCs for nearly all systems in the HOSC
 - ◆ Tier two devices act to decouple users from the application servers





Implementation

EHS Tier One

- This client architecture is acceptable for users
 - ♦ Client security is an issue for remote users
 - ♦ Network performance may seriously degrade X-windows
 - ♦ Management of the clients will not be by the EHS SMAC capability
- However more was needed to be done to provide a more robust environment and make these services available for remote users at the client level
 - ♦ The HOSC looked to technology in the form of communication enhancements
 - Firewall and VPN technology
 - Level 1 and 2 network technology
 - ♦ The HOSC web services had long been supported by SSL and a PKI infrastructure
 - That level of security was needed for remote users
 - However the HOSC utilized 4 products, 2 quite expensive, to secure sessions
 - An alternative was found in VPN technology which could be used to bundle multiple protocols over one virtual pipe including web, X-windows, telnet, and the custom data packets which source the native EPC applications



Implementation

EHS Tier One

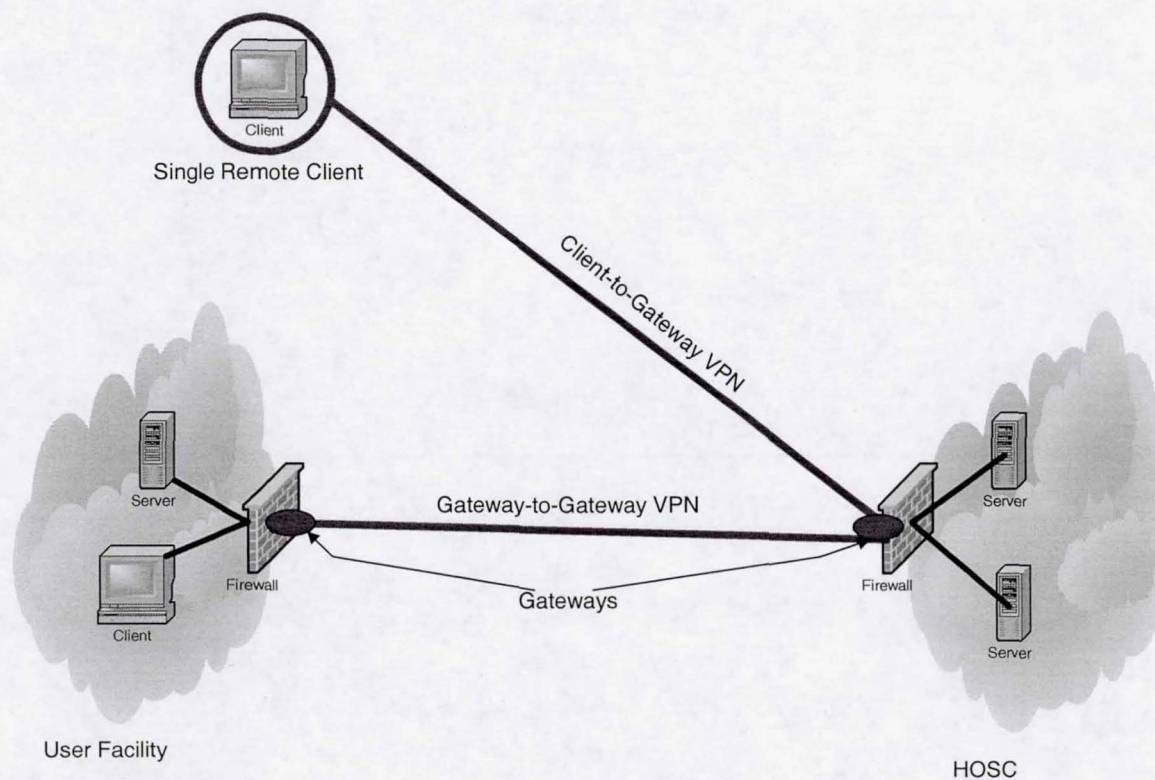
- The HOSC adopted VPN as a standards based technology to replace all of its secure connections
 - ◆ Additionally the HOSC firewall had become a victim of the TELCO wars
 - The HOSC was evaluating new firewalls which were more flexible
 - The selected HOSC firewall supports IPSec compliant VPNs
 - ◆ Therefore the HOSC selected a firewall to provide a secure environment and provide an encrypted gateway for all users of the HOSC
 - Gateway to Gateway VPNs are supported for facilities
 - Gateway to Client VPS are available for individual users
 - A reduction in managed protocols through firewalls reduces problem investigation by O&M personnel



Implementation

EHS Tier One

HOSC VPN Implementation





Implementation

EHS Tier One

- To provide a more robust environment for local users, network technology was investigated
 - ♦ The HOSC was networked with shared media gear, FDDI and ethernet
 - ♦ When support was found it was very expensive for sunset technology; e.g. FDDI
 - ♦ Therefore it was necessary as well to migrate the HOSC
 - Fortunately the HOSC migration was planning to replace all platforms
 - New, inexpensive, and robust hardware (NICs and switches) was available
 - Reliability numbers were far in excess of previous hardware
 - ♦ To migrate the HOSC, VLAN technology is employed
 - External networks were transitioned first
 - Internal FDDI LANs were bridged to new switches with routing capability
 - VLANs were created which allowed migration of hardware to Fast-E without disruption
 - Few systems had reliance on a large MTU or hardcoded to use a FDDI NIC
 - Bonuses from IP switching not only include lower cost but also
 - Full duplex, point to point, fast-E
 - Increased security by not allowing promiscuous network access



Implementation

EHS Tier One

- System management of the PCs was not provided by custom code therefore Windows 2000 Advanced server was used
 - ♦ The EHS management system (SMAC) was not extended out to the PC environment
 - ♦ The Window 2000 server was found to be a robust substitute
 - ♦ PC Servers have been procured to administer/support local PCs (EPC)
 - LDAP, Domain Controllers, File Servers, Print Servers, etc.
 - Microsoft Software Installation Extension
 - Utilization of group policies and Active Directory
 - ♦ Allows centralized management
 - ♦ What does it support
 - Initial software deploy
 - Mandatory and non-mandatory upgrades
 - Applications
 - O/S patches
 - Application patches
 - Software removal



Implementation

EHS Tier Two

- Tier two is the interface domain
 - ♦ Interface servers validate user request and act as gateway to HOSC services
 - ♦ No persistent data is present
 - ♦ Interface servers are sacrificial
- Tier two is composed of three types of servers
 - ♦ Web servers proxy non-realtime services
 - ♦ RIS (Remote Interface Servers) proxy command interfaces and support UDP delivery of telemetry
 - ♦ Login servers act as X-clients and data servers for PC clients
 - ♦ These servers are protected by a firewall with stateful packet filtering
 - Access is over secure network interfaces
 - Users are restricted based on the users' IP address
 - ♦ Many users are supported by each server type



Implementation

EHS Tier Two

- Tier two is composed of three types of servers
 - ♦ Tier two dependencies with tier one are at the protocol level and can interoperate with any type of appropriately configured tier one platform, whether local or remote
 - Tier two platforms interface with EHS PCs
 - Tier two platforms interface with TReK PCs
 - Most International Partners and some CONUS facilities utilize locally developed systems
 - ♦ Tier two platforms are being migrated to Linux



Implementation

EHS Tier Three

- Tier three is the application domain
 - ◆ Critical data to include user security information is persistent
 - ◆ These servers are protected at all cost
 - Users have privileges checked to access
 - Input data is scanned for viruses, trojan horses, worms, etc.
 - Many checks overlap to ensure complete coverage
 - ◆ Critical mission functions are performed
 - ◆ These servers encompass a broad range of services to include
 - ISS drop boxes
 - Telemetry archive
 - Project database
 - Telemetry and science data distribution
 - Long and short term plan
 - Commanding
 - Exception monitoring
 - Payload Information Management System
 - ◆ These systems are considered inviolate and protected at all cost
 - Systems here include PPS, PDSS, and EHS
 - At least five (5) operating systems and hardware platforms are encompassed



Implementation

EHS Tier Three

- Tier three is the application domain
 - ◆ Servers which host components from other facilities will not be migrated
 - ◆ All other platforms will be migrating to Linux, to include tier two
 - Infrastructure servers such as DNS and Sendmail
 - Application servers and interface servers
 - LDAP will be utilized for common security items
 - ◆ As stated earlier, new servers will be configured to the VLANs



Implementation

Hardware Platforms

- Whereas the HOSC has been supporting multiple vendor platforms, the use of the Linux O/S allows for a new choice for platform
 - ◆ Many proprietary architectures are expensive and make dubious performance claims
 - ◆ The HOSC performed a number of benchmark studies and eventually generated three (3) prototype studies with client and server-side applications
 - ◆ Results were remarkable
 - A 2.4 GHz Dual Intel Pentium Platform can compete with RISC processors that are eight times more expensive
 - Spec2000 shows comparable compute capability with much more expensive platforms in almost all categories even comparing 32 to 64 bit processing
 - Our prototypes include nearly all of our COTS products and most of our applications
 - Opens source and high performance proprietary data bases
 - Web, Java, and X-windows all performed comparably or better
 - High performance multi-threaded applications were well behaved



Implementation

Hardware Platforms

- Whereas the HOSC has been supporting multiple vendor platforms, the use of the Linux O/S allows for a new choice of platform
 - ♦ One of our prototypes was generated to understand portability and development issues
 - The prototype looked at over two million lines of code
 - Was conducted by experience programmers without specific domain expertise
 - Software normally executes on two Unix platform vendors
 - Found that porting was straightforward
 - There was a learning curve which was not onerous
 - Most tools had idiosyncrasies
 - We uncovered some of our own latent problems due to the rigor applied

At this point we became convinced that Linux is a viable solution and that the use of commodity hardware is a viable solution!



Implementation

Hardware Platforms

- As a sidebar to the HOSC migration activity, the Chandra X-Ray Observatory ground system is an EHS ground system as stated earlier. They have opted to port their entire ground system to Linux on commodity hardware. This includes their workstations and servers.



Implementation

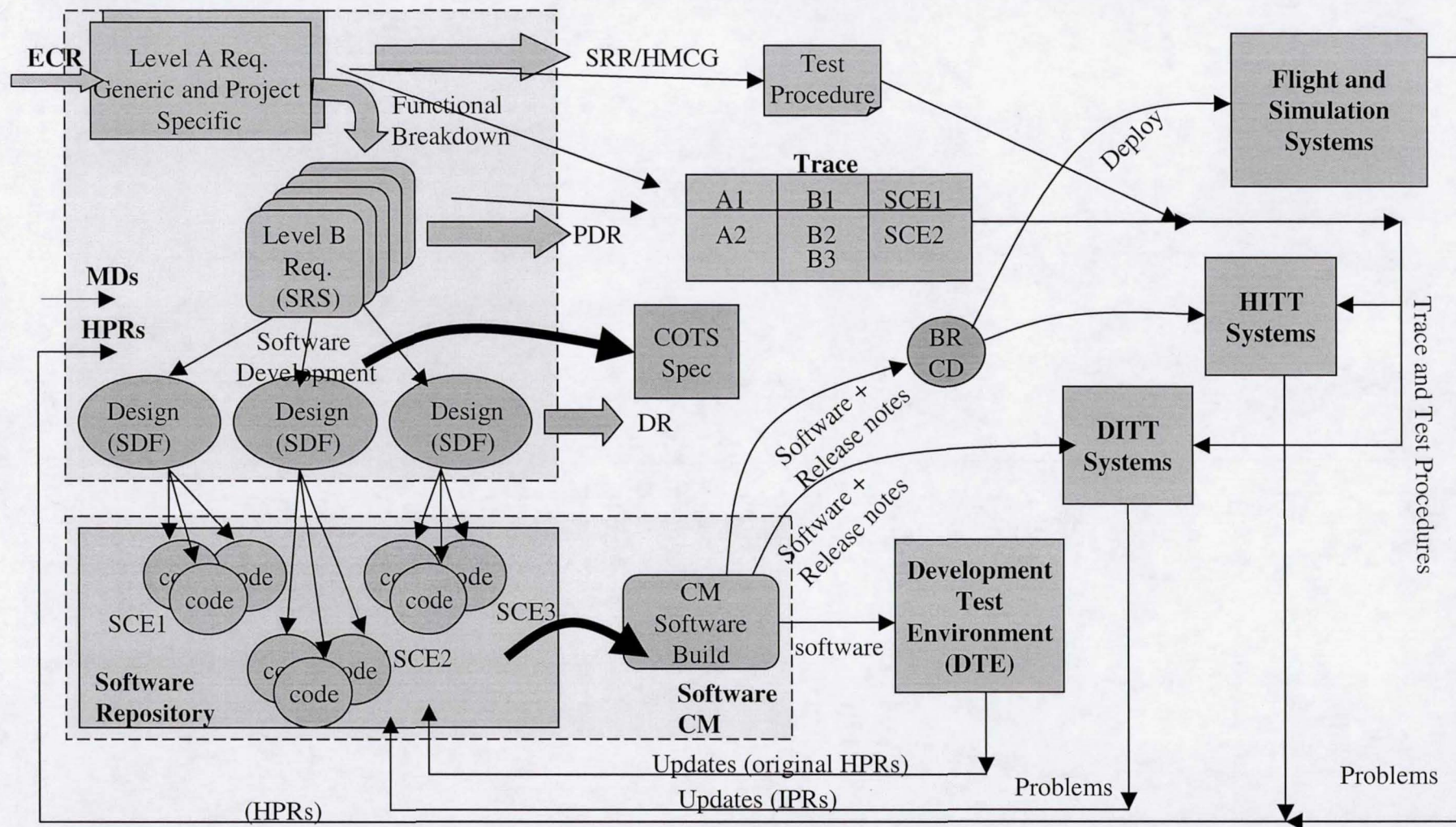
Requirements and software

- Just being able to is not enough, the process must be defined and documented
 - ◆ For the HOSC, this includes a review of the Level A requirements
 - Level A requirements define the overall requirements of the segment
 - Requirements are housed in two documents; general and project specific
 - Requirements are at the now superseded 2167A level standard
 - ◆ Requirements are further decomposed into level B requirements and implementation
 - Software Requirements Specifications (SRS)
 - Software Design Folders (SDF)
 - ◆ Upon review of the A level specifications it became apparent that limitations were inherent
 - Limitations inhibited the deployment of new technology
 - Limitations inhibited the development of new methods
 - ◆ As a result, the Level A documents went under revision
 - Requirements were rewritten for simplification and to remove implementation
 - Requirements were rewritten to allow the use of new techniques
 - Most document structure was maintained to promote easy traceability
-



Implementation

Requirements and software





Implementation

Requirements and software

- The result of the requirements rewrite was the use of several new technologies
 - ◆ First, it became apparent that operations could be improved by consolidating storage
 - Data could be made available to any platform through a Fibre switched architecture
 - Servers would no longer have to have unique physical configurations
 - When coupled with the switched network architecture, servers can effectively be banked
 - ◆ Second, new tools could be utilized by development and in operations which enhanced portage and re-engineering of capabilities
 - Use of new languages and techniques
 - Ability to remove expensive and unique COTS and replace by more generic and common products



Implementation

Requirements and software

- The HOSC would implement a highly available storage architecture
 - ◆ NAS Implementation
 - Administration services (COTS SW Configuration)
 - Mission operations where performance is not an issue
 - ◆ SAN Implementation
 - Database related
 - High performance solutions
 - ◆ The architecture would be made available to all hosted systems; PPS, EHS; PDSS
 - ◆ Coupled with reduced commodity hardware cost, operations can be more flexible
 - All boxes would have access to SAN/NAS storage
 - Any box can support any operational function
 - Software versions and O/S images can be distributed fast and securely



Implementation

Requirements and software

- Expanding the tools set opened up a whole new range of possibilities
 - ♦ Support for NASA direction of faster, better, cheaper with manageable risk
 - ♦ Windows development was a pleasant surprise
 - Tools selection to support Windows 2000 development were more varied
 - Cost for tools was lower than Unix and usually appear before or with Solaris and Linux variants
 - Training was easy to come by
 - For experienced developers with domain knowledge, productivity is high
 - ♦ For the UNIX diehards, Linux was an interesting development
 - Most COTS products used by the HOSC were available
 - Intriguing open source substitutes were available
 - For experienced developers with domain knowledge, productivity is high
 - The price is right
 - ♦ Several other technologies we were using also helped us to integrate the heterogeneous environment; e.g. LDAP and JAVA



Implementation

Requirements and software

- A brief word on security in the Windows environment
- Fact vs. Fiction: The debate over Unix vs. 2000 security is based largely on “religion” rather the inherent security designed into the Operating Systems
- Fictions
 - ◆ Unix is more secure than Windows 2000
 - ◆ Windows 2000 is more secure than Unix
- Fact
 - ◆ There is no “universal truth” regarding the relative security of Unix vs. 2000 - arguments exist to support both positions
 - ◆ PCs are generally less secure ONLY because most aren’t professionally administered
 - ◆ The “body of knowledge” for securely configuring Windows 2000 has been developed commensurate with 2000’s increasing role in security-sensitive applications

The HOSC has a layered security environment and the O/S controls are merely one layer. The environment is administered by professionals.



Implementation

Requirements and software

- Level B requirements are the decomposed level A requirements in terms that can be managed by a relatively small number of developers
 - ◆ The requirements should be written with sufficient detail that the development team can implement
 - ◆ Every A level traces to Level B requirements
 - ◆ Even though the process can be somewhat subjective
 - Ensures no level A requirement is left unimplemented
 - Ensures spurious requirements are not implemented
- Level B requirements are review by relevant parties during design reviews for completeness
 - ◆ Custom code is defined in this fashion
 - ◆ COTS products are generally identified in this way only when they are tightly coupled
 - ◆ COTS products per se are captured in a separate COTS specification



Implementation

Requirements and software

- Software design begins once SRSs are approved
 - ◆ Design is captured in the Software Development Folders (SDF)
 - ◆ SDFs contain whatever documentation is required to develop the software
 - Functional data flows
 - Descriptions of interface to other CSCIs
 - Software architecture drawings
 - Most other information is captured in the source
- The developers generate code, begin testing, and create our unique system based on a well defined environment
 - ◆ Naming standard and coding conventions
 - ◆ POSIX compliance
 - ◆ Well defined configuration model supported by a complete CM tool
 - ◆ Test environment which mimics but is not a copy of the flight systems



Implementation

Requirements and software

- The HOSC software progresses through 3 successive test environments
 - ◆ Development Test Environment
 - Configuration managed software is tested by developers
 - Software builds are updated often
 - Test platforms are similar to operations and interfaces are verified
 - ◆ Development Integration Test
 - Configuration managed software is tested by a test team
 - Test platforms approach operations in capability and layout
 - New capabilities are verified at a system wide level
 - Initial regression testing is accomplished
 - ◆ HOSC Integration Test
 - Configuration managed software is tested by a test team
 - Test platforms are identical to operations
 - Full regression testing is accomplished
 - Product is certified for Simulation and Flight



Implementation

Requirements and software

- The HOSC has had a traditional development and deployment cycle
- With migration of the HOSC systems, the model had to be modified
 - ♦ It was wasteful of scarce resources
 - ♦ Did not allow deployment of capabilities as they became available
 - ♦ Did not allow for much user feedback during product development
 - ♦ Would not have allowed us to do better as opportunities presented themselves
- Therefore the HOSC adopted a spiral development and deployment philosophy
 - ♦ This was possible because the basic needs had not changed
 - ♦ Algorithmic processing had not changed
 - ♦ We wanted the user to be involved throughout the migrations
 - ♦ We could reap savings as opportunities presented themselves, e.g. replacement of COTS products, moving processing between platforms, etc.



Initiatives



Initiatives: What We Looked For

- Support Issues
 - ◆ EOL on platforms
 - ◆ Vendor viability
 - ◆ Slow or waning support for COTS
 - ◆ Increasing maintenance costs
 - ◆ Increasing COTS costs
- Performance Issues
 - ◆ Platforms can no longer be upgraded
 - ◆ Increased failure rates
 - ◆ Expanding user base
 - ◆ Evolving/expanding requirements
 - ◆ Individual users need for more screen real estate and performance



Initiatives: What We Looked For

- O&M Manpower Issues
 - ◆ Increased manpower required to maintain systems
 - ◆ Too many different hardware vendors and operating systems and COTS packages
 - ◆ Increased manpower required to operate the system
 - ◆ Reduced budgets
- COTS Issues
 - ◆ No longer used or underutilized COTS packages
 - ◆ COTS isolation
 - ◆ Unique, site specific COTS
 - ◆ Changing COTS versions
 - ◆ Multiple COTS packages performing similar functions



Initiatives: What We Looked For

- General Issues
 - ◆ Obsolete capabilities
 - ◆ Multiple EHS applications performing similar functions (like telemetry storage)
 - ◆ Technology insertion
 - ◆ Commodity priced components
 - ◆ Consolidation
 - ◆ Security



Initiatives: PC Migration

- The HOSC had approximately 130 UNIX workstations in Operations, development and testing that reached end of life (EOL) in June 2002
- The workstations had the maximum allowable memory (256MB), fastest available CPU's, and some were as much as 8 years old
- The primary focus of PC Migration is to move critical Cadre end-user applications from expensive UNIX workstations to lower cost Windows 2000 PC platforms
 - ♦ EHS PC (EPC) acts as an X-Window server with the workstation software (legacy) running on a UNIX server (RIS/X-Windows Server)
- Advantages
 - ♦ Fewer different types of desktop platforms in facility to maintain, test and certify
 - Now all end user client platforms are W2K PCs
 - ♦ Significantly cheaper than replacing the EOL UNIX workstations with new UNIX workstations (both initial purchase and recurring license costs)



Initiatives: PC Migration

- Advantages
 - ◆ RIS/X-Window Servers provide more efficient usage of expensive UNIX CPUs/applications (through shared CPU/memory/disk vs. individual workstations)
 - Base EHS applications run only once (e.g., telemetry acquisition/distribution tasks)
 - Greater expandability options for CPUs and memory that benefit more than one user
 - The RIS/X-Window Server only performs the “x-client” function
 - The “X-server” function is offloaded to the PC, which has significantly better graphics performance and response
 - ◆ Reduced reconfiguration times and complexity (i.e., ~ 3-4 servers to reconfigure rather than 40+ workstations for a support activity such as flight)
 - ◆ Improves system administration activities
 - Through the use of Windows 2000 tools for desktop/enterprise management
 - ◆ New console hardware can be immediately deployed to operational areas, allowing realization of faster non-labor cost reductions to help offset short-term software development labor increases



Initiatives: PC Migration

- Advantages
 - ◆ EPC desktop advantages
 - Provides low-cost capability for multiple CPUs and display monitors (for improved Cadre task automation/execution), vs. equivalent UNIX W/S implementation
 - Available with Fast-E (Ethernet) network interface at minimal cost; eliminates “sun setting” Fiber Distributed Data Interface (FDDI) network interface and hardware
 - PCs priced at commodity level; large number of vendors
 - Can take advantage of Moore’s Law
 - Provides more user-friendly desktop to Cadre user, including standard COTS tools (Microsoft Word, Excel, Access, etc.), which simplifies training
 - PCs support EHS developed Web/JAVA applications, which were not functional on the UNIX workstations
 - ◆ Performance of end-user graphical applications (e.g. Display Ops) improved dramatically
 - Prototypes performed showed significantly better performance than versions running on the UNIX workstations



Initiatives: PC Migration

- Advantages
 - ◆ PCs have more/better development tools which increases development and maintenance productivity
 - ◆ Improved remote access to Cadre products
 - This architecture allows EPCs to be remote to the HOSC and view the same displays that the Cadre is viewing
 - ◆ Allows for the reuse of previously developed Cadre products
- Disadvantages
 - ◆ Required intensive software development investment
 - ◆ Increased risk of software defects due to the redesign and rewrite of certified EHS applications onto the PC platform
 - ◆ Moves toward a "Microsoft-centric" world (versus the current POSIX, X-Windows, JAVA portability baseline)
 - ◆ A single RIS/X-Window server failure will affect multiple users (as compared to the single workstation architecture)



Initiatives: PC Migration

- High performance and high throughput client-side applications will be re-written to run on EPC (native)
 - ♦ Will significantly offset the load on UNIX servers
- The deployment strategy provides for support of legacy HOSC X-windows applications in parallel with new Native PC services until each new application is completely certified by the Cadre
 - ♦ Provides Cadre fallback position if problems experienced with new applications
- PC Migration is being developed in multiple phases/releases
 - ♦ Phase 1.0 deployed EPCs in place of operational workstations (completed)
 - ♦ Phase 2.0 focused on migrating the initial set of X intensive applications to native Windows 2000 PC architecture (completed)
 - ♦ Phase 3.0 primarily focuses on moving the rest of the X intensive applications to native Windows 2000 PC architecture (on schedule 7/03)
 - ♦ Phase 4.0 focuses on moving the “generation” applications to EPC (on schedule 1/04)



Initiatives: PIMS Redesign

- Change Overview/Description
 - ◆ Significant recurring cost reduction was achieved by removing a commercial off the shelf (COTS) product as PIMS workflow engine (replaced with custom developed code)
 - Only a small portion of the COTS product was actually used by PIMS
 - The developed workflow engine was written to meet the PIMS workflow requirements
 - ◆ Included several workflow related Engineering Change Requests (ECRs), HOSC Problem Reports (HPRs), and additional customer feedback comments (from Increment 2 users)
- Advantages
 - ◆ Eliminated largest recurring COTS software maintenance cost in EHS (>\$300K per year, escalated over 5 years to >\$500K/year)
 - ◆ Simplified PIMS server architecture; eliminates many failure points, processes, scripts, and data



Initiatives: PIMS Redesign

- Advantages
 - ♦ Eliminated costs and risks to keep workflow engine and other COTS compatible with the operating system
 - ♦ Improved software transaction control, equating to better data integrity
 - ♦ Several related ECRs, HPRs, and other improvements rolled in
 - ♦ One year of savings in COTS costs paid for the labor to replace it
- Disadvantages
 - ♦ The development effort limited the ability of the PIMS team to work several other ECRs during the implementation timeframe (about 6 months)



Initiatives: PDSS Consolidation

- Change Overview/Description
 - ◆ Consolidate Payload Data Services System (PDSS) packet processing, data distribution and data storage functionality onto single platform to reduce operational complexity and cost
 - ◆ Project Objectives
 - Reduce the recurring vendor licensing, support, and maintenance fees
 - Reduce the number of system elements to be configured, monitored, and maintained
 - Reduce the number of software lines of code
 - Simplify the Operations interface and system configuration
 - Provide auto fail over capabilities for real time operations
 - Position the system for future data rate increases (150Mbps)



Initiatives: PDSS Consolidation

- Advantages
 - ◆ Development performed by existing PDSS team
 - ◆ Design supports direct applicability into 150Mbps upgrade plans, reducing cost of 150 Mbps upgrade for payloads
 - ◆ Significant cost savings generated by reduction in annual hardware vendor maintenance
 - ◆ Simplified PDSS facility operations by having one PDSS Distribution Server per activity (Flight, Test, Sim)
 - ◆ Auto failover capability for flight support
 - ◆ Allows end-user to control own real-time destination data routing (instead of PDSS Operator)
 - ◆ Phased approach minimizing risks vs. replacing entire system at one time
 - ◆ PDSS Server Consolidation estimated to save program over \$1M in maintenance and labor over 5 years
 - Disadvantages
 - ◆ Additional labor needed for FY02 & FY03 to develop software
 - ◆ Investments in hardware upgrades
-



Initiatives: PDSS Consolidation

- Data Distribution and Storage – Phase I
 - ♦ Eliminate 12 production PDSS Data Distribution Enterprise Server class machines to be replaced with Intel/Linux PC workstation class machines
 - ♦ Consolidate multiple PDSS processes into a single, multi-threaded process which will perform limited packet processing and all Data Distribution and Storage functions (PDSM)
- Front-End Processors (FEPs) – Phase II
 - ♦ Implement alternative, less-expensive FEP system (supporting at least 150 Mbps) (+4 systems)
 - ♦ Decommission/retire current FEP system (TSI Telsys) hardware/software (- 8 systems)
 - ♦ Eliminate Asynchronous Transfer Mode (ATM) switches (- 2 switches)
 - ♦ Add full packet processing capabilities to the PDSM server process with auto-fail over capabilities



Initiatives: Server Migration

- Change Overview/Description
 - ◆ Server Consolidation study task
 - Avoid significantly higher refurbishment costs
 - One-for-one replacement on all platforms was not feasible with the current budget
 - Identify additional cost reductions that can be realized
 - ◆ Major Considerations
 - UNIX servers reach product end-of-life (EOL) in 2003
 - The RDBMS vendor software support on UNIX servers disappears (January 2006)
 - Potential RDBMS database pricing structure increases
 - ◆ Study Phases
 - Study Analysis 1: Analysis of a consolidated UNIX server architecture
 - Study Analysis 2: Due to concerns with the UNIX server vendors corporate viability and COTS vendor support, an additional study was initiated to review alternate EHS server vendor solutions



Initiatives: Server Migration

- The Option chosen was to port EHS software from UNIX servers to low-cost Linux OS/Intel hardware vendor platform and include some consolidation
 - ♦ Platforms involve little to no direct end-user interaction
 - ♦ Includes replacement of 30+ aging UNIX servers
 - ♦ Allows the server functions to be consolidated to a single platform to support more efficient development and testing
- Advantages
 - ♦ Greatly improves deployment and operations costs of services to KSC PTCS POIC, and any potential future "POIC Copy" remote user sites
 - ♦ Significantly cheaper than replacing UNIX servers one-for-one
 - ♦ Provides greater flexibility in test and operational utilization of EHS servers
 - ♦ Allows stepwise enhancements where feasible
- Disadvantages
 - ♦ Requires additional software development investment to perform consolidation as well as the migration from UNIX to Linux



Initiatives: Server Migration

- The study also identified several supporting initiatives
 - ♦ Implement High-Availability Centralized Storage
 - Minimizes overall disk storage requirements and data replication
 - Decreases system configuration, reconfiguration and administration requirements
 - Provides simpler, more flexible, failovers and flight-to-flight transitions
 - Storage for virtually all types of data; user products, Databases, COTS products, O/S images; short term telemetry data
 - ♦ Network Improvements
 - Continue transition of systems from “sunset” FDDI network technology to FastE/GigE
 - ♦ Banking of servers with no persistent data and dynamic load (to support growing user base)
 - As demand increases servers can be easily added to support additional load
 - Involves load sharing among multiple primes with common shared backup



Initiatives: Server Migration

- The study also identified several supporting initiatives
 - ◆ Firewall and Security improvements
 - Integrate Mission Admin System and Enhanced HOSC System (EHS) firewalls and implement load sharing design (nearly completed)
 - Consolidate secure access methods, replace with Virtual Private Network (VPN) technology (complete)
 - Eliminates two technologies and 4 COTS products
 - ◆ Internet Voice Distribution System (IVoDS)
 - Currently IVoDS is in parallel operational testing support phase
 - Appears to be a low-cost alternative to traditional voice
 - Supports our small remote users



Conclusions



Conclusions

- Identify areas with high value targets
 - ◆ Look at operations from the system level
 - ◆ Areas of near-term payoff are always good, however
 - Coupling with other items can have long term benefits
 - Don't discount O&M benefits since recurring cost savings can add up
 - This is especially true for multi-year or multi-project programs
 - ◆ Perform an initial cost/benefit analysis
 - Look for low-hanging fruit
 - Be conservative, there are many unknowns; good and bad
 - Develop a working hypothesis for each opportunity



Conclusions

- Prioritize opportunities and evaluate in relation to the overall system
 - ◆ Never do anything for only one reason, consider mitigating factors
 - ◆ Hit high value targets first, use the saving to pay for other items
 - ◆ Analyze the opportunities as a return on investment (ROI)
 - Immediate Return (We eliminated unnecessary COTS)
 - Long Term Maintenance (Implemented a commodity based architecture)
 - Reduces recurring cost (Reduced operational complexity reduces the need to increase staffing as the mission user base expands)
 - ◆ Review other items as relates to the ongoing operations
 - Performance increases may be required in the near term
 - Expansion of the current baseline may require new capabilities
 - Increased Operational Flexibility
 - ◆ Gain initial approval to proceed on high value targets



Conclusions

- Opportunities must be clearly articulated to management
 - ◆ A unique case was built for the HOSC based on the HOSC situation
 - ◆ Application code was developed to encompass general operational capabilities
 - ◆ Redundant or obsolete capabilities and features have been identified
 - ◆ New capabilities and features can be incorporated and will reduce cost while enhancing operability
 - Highly available storage
 - Fault tolerant systems
 - Consolidation of COTS products which accomplish similar tasks
 - Introduction of high performance network hardware
 - Introduction of CISC servers (Intel type processors)
 - ◆ Get management buy-in and enthusiasm for the proposal
 - Use a detailed proof-of-concept to help build a case
 - Develop a detailed Cost/Benefit analysis



Conclusions

- Once management is in, generate a plan
 - ♦ Set Goals
 - Goals should be aggressive, not insane
 - Final goals will probably not be met, but should be redefined periodically
 - ♦ Develop a migration path
 - Identify key milestones
 - Quantify work and identify ways to measure success (did it cost/save what was predicted)
 - Develop alternatives or strategies for difficult areas
 - ♦ Implement Risk Management activities
 - Educate implementers and users, articulate a vision
 - Develop a detailed schedule
 - Develop a procurement plan
 - Get user feedback where possible to ensure the opportunities are viable in the user community
 - Drag along diehards by leveraging their experience



Conclusions

- Remember
 - ♦ Start with the high value targets and make changes incrementally
 - ♦ Give commodity priced platforms more than a cursory look
 - Use rapid prototyping to prove concept
 - Take advantage of Moore's Law
 - Significantly reduce replacement and maintenance costs
 - ♦ Stay current by utilizing technology insertion
 - ♦ Reduce dependency on expensive, under-utilized COTS
 - ♦ Looks for ways to migrate while maintaining access to legacy systems
 - ♦ Don't try to do it all at once